Robust Graph SLAM in Dynamic Environments with Moving Landmarks

Lingzhu Xiang[1] Zhile Ren[1] Mengrui Ni[1] Odest Chadwicke Jenkins[1]

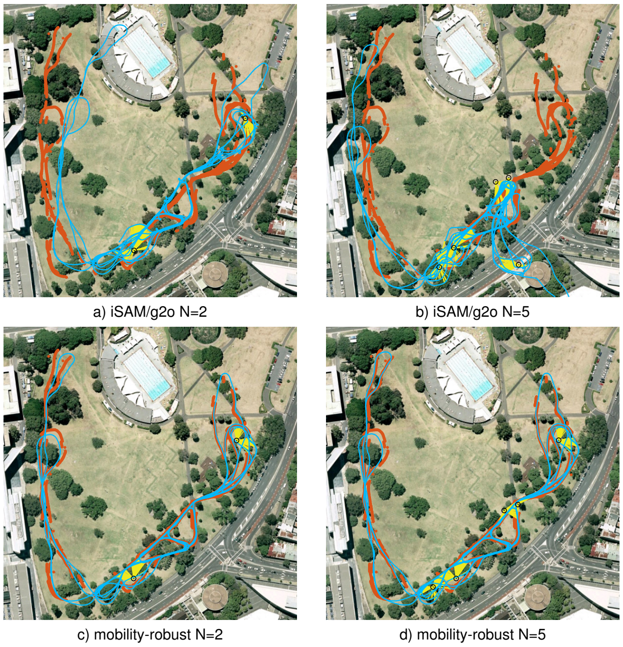
***Abstract*— Recent developments in human-robot interaction brings about higher requirements for robot navigation. Existing Simultaneous Localization and Mapping (SLAM) algorithms face open challenges for navigation in complex dynamic environments due to presumptions of static environments or exceeding computational limitations. In this paper, we propose a robust graph SLAM formuation exploring Expectation Maximization algorithms to characterize landmark mobility while establishing the estimations of robot trajectory and the map. We evaluate the performance of existing robust SLAM algorithms as baselines, and validate the improvement of our new framework against datasets of dynamic environments with moving landmarks.**

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is a central problem for autonomous robots to navigate and perform mobile manipulation tasks. Graph SLAM formulates it as an inference problem on a factor graph. In the factor graph, spatial measurements are the observed factor nodes as constraints between variable nodes of landmark locations and robot poses. The goal of the inference problem is to obtain the maximum likelihood estimate of the joint probability of the graph, which becomes the geometrically consistent estimate of the robot's trajectory and the map. This maximum likelihood estimate on factor graphs can be solved by belief propagation [15], or more recently, by numerical methods after converting into a nonlinear squares [4], [5], [8].

Problems arise in SLAM loop closing when factors incorrectly link unrelated variable nodes. Such false positives effectively create wormholes between spatially distant locations and, thus, collapse and distort the map geometry. In pose graphs, loop closures specify spatial connections between arbitrary poses representing previously visited locations. However, front-end filtering systems often wrongly connect random poses as loop closures. In landmark-based graphs, the data association process can also easily obtain wrong visual feature correspondences, connecting landmarks to the wrong poses. This necessitates robust approaches to SLAM.

SLAM also faces considerable open challenges for dynamic environments. Typical SLAM techniques in the literature are designed for unmanned navigation in uninhabited areas, thus mostly assume static environments with stationary landmarks and loop closures. However, recent human-robot interaction research has seen more applications of navigation in populated, crowded, or social environments where people and furniture moving around is the major characteristics. If the landmarks are moving, the current localization is either kidnapped by the movement, or resulting in distorted maps.


a) iSAM/g2o N=2

b) iSAM/g2o N=5

c) mobility-robust N=2

d) mobility-robust N=5

Fig. 1: Estimating the map and trajectory (blue) on the Victoria Park dataset [4] given landmarks corrupted by simulated movement (black circle) of associated observations (yellow) and GPS ground truth (red). (Top row) the estimation obtained by conventional graph optimization given N=2 (left) and N=5 (right) landmarks corrupted by movement. (Bottom row) the result of mobility-robust map estimation of the same data using our proposed method. Note that stationary landmarks are not shown, and GPS is unavailable at several locations.

In this paper, we present an approach to minimize the effects of moving landmarks, by treating them as outliers, in graph-based SLAM to improve the effectiveness of SLAM in dynamic environments. We propose a mobility-robustified SLAM model that includes a mobility variable over landmarks in the joint probability to scale the effect of a landmark in relation to how stationary it is in space. This mobility variable relates the belief of landmark positions with their measurements, and diminishes moving outlier landmarks within a measurement function (equation 4). Effectively, this mobility variables scale the covariances of Gaussian distributions to loosen the constraints of mobile landmarks such that their motion will push this distribution towards uniformity. An EM-based algorithm, as well as an incremental version, is proposed to infer the mobility scaling and estimate the pose trajectory of a robot with respect to a mobility-robustified objective function. Results of our mobility-robustified SLAM

[1] Lingzhu Xiang, Zhile Ren, Mengrui Ni, and Odest Chadwicke Jenkins are with the Department of Computer Science, Brown University, Providence, RI 02912 [xlz|ren|mni|cjenkins]@cs.brown.edu

are shown respect to the Victoria Park [4] and Alcazar of Seville [10] datasets.

II. RELATED WORK

Graph SLAM has multiple highly efficient optimization solutions. iSAM [4] converts the graph SLAM maximum likelihood estimate into a non-linear least squares optimization problem. The factor graph is incrementally solved by numerical methods, obtaining real-time performance and Bayesian smoothing accuracy. These optimization techniques show the effectiveness of the factor graph formulation of the SLAM problem, and we base our formulation on similar formulations.

A known solution to SLAM in dynamic environments is to maintain two occupancy maps modeling the dynamic and static parts of the environment [14]. By differentiating dynamic and static parts of the environment with different representation, this method is capable of mapping and localization in dynamic environments over time. An alternative approach of dealing with moving objects in dynamic environments is combining SLAM with object detection and tracking. Wang et al. [13] proposed a Bayesian framework to solve the SLAM together with object motion modeling by sophisticated object detection and tracking and data association algorithms. In this approach, object detection and tracking is used as a preprocessing front-end to filter out moving objects.

Robust SLAM techniques have been proposed to solve front-end outlier problems without relying on pre-filtering. Some use robust objective functions or robust representation of observations. Dynamic Covariance Scaling [1] adds a robust kernel factor to regularize the Mahalanobis errors in the Gaussian distributions of landmark observations. Max-Mixture [7] enhances factor potentials with a clever representation for mixtures of Gaussians in place of a unimodal Gaussian distribution. This kind of approaches still assume sources of errors being mostly perceptual aliasing in wrong loop closures, without regard to environmental movement. Unless modeled explicitly in factor potentials, these methods will have difficulty in handling the movement of landmarks.

Front-end outliers and dynamic elements can also be addressed through identifying mobility as part of the back-end graph optimization framework. Haehnel et al. [3] and Rogers et al. [11] extend graphical model formulations with a latent indicator variable to infer whether a landmark is mobile. EM algorithms are used to iteratively infer these latent landmark mobility variables in the graphical model and estimate the optimal SLAM solution. The switchable constraints [12] approach allows the optimizer to naturally change the topological structure of the problem during the optimization itself using switch variables as a multiplicative scaling factor on the information matrix associated with that constraint. However, these EM based algorithms lack the robustness provided by previous techniques. Further, an observation-based indicator only models the observation it associates with and will not characterize the mobility of landmarks which associate with multiple observations.

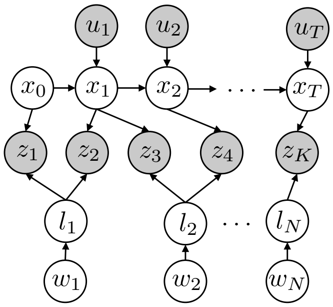


Fig. 2: Graphical Model Formulation. Dark nodes are observed.

III. MODEL: AUGMENTED GRAPH SLAM

Following [4] we formulate the SLAM problem in graphical models as in Figure 2. Specifically, the robot states (as position and orientation over time in map coordinates) are denoted by $X = \{\boldsymbol{x}_i\}$ with $i \in 0, \ldots T$, the landmark locations in map coordinates by $L = \{\boldsymbol{l}_j\}$ with $j \in 1, \ldots, N$, the control inputs for movement by $U = \{\boldsymbol{u}_i\}$ for $i \in 1, \ldots, T$ and the landmark measurements in robot coordinates by $Z = \{\boldsymbol{z}_k\}$ with $k \in 1, \ldots, K$. In addition to the classical graph SLAM formulation, we augment the representation of landmarks with a set of N scalar latent parameters $W = \{w_j\}$ with $j \in 1, \ldots, N$. At each time k, the measurement $\boldsymbol{z}_k$ corresponds to robot pose $\boldsymbol{x}_{i_k}$, landmark $\boldsymbol{l}_{j_k}$, and latent variable w_{j_k}, where i_k associates robot poses with measurements, and j_k associates landmarks with measurements.

In considering dynamic environments, W models the mobility of each landmark about whether it is capable of movement or not without considering extra kinematics, and robustify the observation term of the model in Equation 4. Through W, corrupted measurements associated with moving landmarks are suitably eliminated as outliers from the mapping process. It may be appealing to model a dynamically moving object as a sequence of variables. However, it is shown in the following that the scalar mobility variables are adequate enough to eliminate moving landmarks from graph optimization.

According to the proposed graphical model, we give the joint probability of all variables and measurements as:

$$P(X, L, U, Z, W) \propto \prod_i P(\boldsymbol{x}_i | \boldsymbol{x}_{i-1}, \boldsymbol{u}_i) \prod_k P(\boldsymbol{z}_k | \boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}, w_{j_k}). \quad (1)$$

Then the maximum likelihood (ML) estimate of the unobserved poses X and landmarks L given observations Z, known controls U, and the current latent parameters W are defined as

$$X^*, L^* = \arg\max_{X,L} P(X, L, U, Z, W). \quad (2)$$

To calculate the ML estimate, the objective is linearized and converted into a linear least squares problem in this form $\arg\min_{\boldsymbol{\delta}} ||\boldsymbol{A\delta} - \boldsymbol{b}||^2$ by algebraic manipulation, and then optimized using different numerical methods. Derivation with more details is provided in appendix A.

Using a Gaussian representation with the latent extension, the sensor model, the process model and measurement equation are defined as

$$\begin{aligned} \boldsymbol{x}_i &= f_i(\boldsymbol{x}_{i-1}, \boldsymbol{u}_i) + \boldsymbol{\eta}_i \\ \boldsymbol{z}_k &= h_k(\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}) + \boldsymbol{\theta}_k \end{aligned} \tag{3}$$

where $\boldsymbol{\eta}_i$ and $\boldsymbol{\theta}_k$ are noise terms which follow zero-mean Gaussian distribution with covariance matrices $\boldsymbol{\Gamma}_i$ and $\boldsymbol{\Sigma}_k$. With this formulation, the second part of the joint probability 1 is augmented with the mobility indicator. In addition we also apply a robust kernel v_k to the observation term, then the conditional probability of $\boldsymbol{z}_k$ is defined as an augmented Gaussian distribution:

$$\begin{aligned} P(\boldsymbol{z}_k | \boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}, w_{j_k}) \propto \exp(-w_{j_k} \tilde{\boldsymbol{\mu}}_k^T \boldsymbol{\Sigma}_k^{-1} \tilde{\boldsymbol{\mu}}_k), \\ \tilde{\boldsymbol{\mu}}_k = v_k(h_k(\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}) - \boldsymbol{z}_k) \end{aligned} \tag{4}$$

where w_{j_k} represents the likelihood of being static for landmark $\boldsymbol{l}_{j_k}$ associated with measurement $\boldsymbol{z}_k$ at time k, and v_k is the robust scaling factor associated with each landmark observation representing whether the measurement is an inlier. When w_{j_k} or v_k approaches zero, the effect is equivalent to making the covariance of the Gaussian very large, effectively rendering the distribution uniform and the constraint represented by the distribution of no impact on the graph optimization process. Note that w_{j_k} can be negative because the formulation given here is proportional to a normalizing constant.

IV. ALGORITHMS

Following the notation in section III, we introduce the Expectation Maximization (EM) algorithms for estimating w with robustified objective function to learn the mobility of landmarks and estimate the robot trajectory and the map.

A. Estimation of Latent Parameters

As described in [11], in Equation 4, the hidden variables w_k must be estimated from multiple observations of each landmark. In the **M step**, we select the optimal $\boldsymbol{w}$ which maximizes the joint likelihood. However there is a trivial solution to the likelihood maximization which sets $w_k = 0$ for all k, we penalize the log-likelihood objective in Equation 5 with Lagrange multiplier which act as priors of the latent landmark mobility variables:

$$\begin{aligned} &\mathrm{Obj}(Z, X, L, W) = \\ &\quad \sum_k \left[-w_{j_k} (\tilde{\boldsymbol{\mu}}_k^T \boldsymbol{\Sigma}_k^{-1} \tilde{\boldsymbol{\mu}}_k) \right] - \frac{1}{2} \lambda (1 - \boldsymbol{w})^T (1 - \boldsymbol{w}) \end{aligned} \tag{5}$$

where $\tilde{\boldsymbol{\mu}}_k = v_k(h_k(\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}) - \boldsymbol{z}_k)$ is the robustified prediction error of observation k obtained from the E step, and w_{j_k} is the weight to estimate for landmark $l = j_k$ associated with the k-th observation. We equate the derivative of the objective function with respect to w_l to zero, and maximize the log likelihood, then for each w_l we get

$$w_l = 1 - \frac{1}{\lambda} \sum_{k \in K_l} \tilde{\boldsymbol{\mu}}_k^T \boldsymbol{\Sigma}_k^{-1} \tilde{\boldsymbol{\mu}}_k \tag{6}$$

where K_l is the set of measurements of landmark l, and λ is an assigned constant parameter to trade off the penalty. Larger λ will penalize more against the number of landmarks being mobile $w_k = 0$ for any k.

B. Graph SLAM Optimization

In the **E step**, we use the estimated latent parameters to obtain minimum variance results with graph optimization methods. We employ the standard approach of graph optimization as in [5], with our augmentation to the objective by introducing landmark mobility indicators w_{j_k} and observation robust kernel v_k. Specifically, we want to compute the maximum likelihood estimate of the robust objective

$$\begin{aligned} X^*, L^* &= \arg\max_{X,L} \{ -\log p(X, L, U, Z, W) \} \\ &= \arg\max_{X,L} \{ \sum_{i=1}^{M} ||f_i(\boldsymbol{x}_{i-1}, \boldsymbol{u}_i) - \boldsymbol{x}_i||^2_{\boldsymbol{\Gamma}_i} \\ &\qquad + \sum_{k=1}^{K} w_{j_k} ||v_k(h_k(\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}) - \boldsymbol{z}_k)||^2_{\boldsymbol{\Sigma}_k} \\ &\qquad + \sum_{k=1}^{K} ||1 - v_k||^2_{\boldsymbol{\Xi}_k} \} \end{aligned} \tag{7}$$

where the $||.||_{\boldsymbol{\Sigma}}$ term denotes the Mahalanobis distance with covariance $\boldsymbol{\Sigma}$. The penalty term $||1 - v_k||^2_{\boldsymbol{\Xi}_k}$ here is the switching prior introduced in [12]. Note that in this step, $\boldsymbol{w}$ is fixed and the penalty term with respect to w_k in the objective for estimating $\boldsymbol{w}$ becomes a constant, thus it does not appear in the objective function.

Here we apply the Dynamic Covariance Scaling[1] kernel v_k to each individual observation k. The factor v_k for observation k is given as:

$$v_k = \min \left\{ 1, \frac{2\Phi}{\Phi + w_{j_k} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\mu}_k} \right\} \tag{8}$$

where $\boldsymbol{\mu}_k = h_k(\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}) - \boldsymbol{z}_k$ is the original prediction error of observation k. The DCS kernel is shown to result in an upper bound specified by Φ for the second part of the objective function. By specifying the upper bound Φ for all robust edges, v dynamically scales the information matrices of all edges each iteration within the graph optimization process, and replaces the covariances of outliers with large ones making it near uniform. Φ is chosen as 1 in our implementation as suggested by its author.

With some derivation (Appendix section), the optimization problem can be formed as a standard least-squares problem:

$$\boldsymbol{\delta}^* = \arg\max_{\boldsymbol{\delta}} ||\boldsymbol{A}\boldsymbol{\delta} - \boldsymbol{b}||^2 \tag{9}$$

The algorithm, while reliable as shown in experiments in [3], is possible to have variants modified for real-time updates. In our current implementation we perform batch optimization. We need to rerun the EM algorithm if new measurements are observed. Incremental implementation of the framework is subject for future work.

C. Incremental Optimization

We introduce the theoretical basis for an incremental implementation of the EM algorithm for the problem. As suggested in [6], if the joint probability is fully factorized with regards to the examples, as is in our case, we can use the following algorithm for update in the E and M step while preserving the correctness:

a) E step: In equation 7, if we have one new observation $\boldsymbol{z}_{K+1}$ at time $K+1$, then

$$\begin{aligned}\boldsymbol{x}^*_{i_{K+1}}, \boldsymbol{l}^*_{j_{K+1}} = \underset{\boldsymbol{x}_{i_{K+1}}, \boldsymbol{l}_{j_{K+1}}}{\arg\max} \ & \{\sum_i ||f_i(\boldsymbol{x}_{i-1}, \boldsymbol{u}_i) - \boldsymbol{x}_i||^2_{\boldsymbol{\Gamma}_i} \\ & + \sum_{k=1}^{K+1} w_{j_k} ||v_k(h_k(\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}) - \boldsymbol{z}_k)||^2_{\boldsymbol{\Sigma}_k} \\ & + \sum_{k=1}^{K+1} ||1 - v_k||^2_{\boldsymbol{\Xi}_k}\}\end{aligned} \tag{10}$$

and equation 10 can also be written in the form of 9, but with less parameters. So, the incremental E step becomes

- Choose one observation term k to be updated, such as $k = K+1$
- Set $\boldsymbol{x}^{(t)}_{i_p} = \boldsymbol{x}^{(t-1)}_{i_p}, \boldsymbol{l}^{(t)}_{j_p} = \boldsymbol{l}^{(t-1)}_{j_p}$ for $p \neq k$
- Set $\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}$ according to equation 7 in equation 9's form.

b) M step: The same step as introduced in section IV-A, using the derivation in equation 6.

With this incremental EM algorithm, the E step can be much faster since the optimization problem has less parameters. Also, after several iterations, we may use the model to update the parameters by adding more observations, without the need to run the EM algorithm on all the observations again.

V. EXPERIMENTS

We implemented our method as discussed previously and compared the result with various alternative approaches. Our implementation is based on the graph optimization framework g2o [5], and we programmed a plug-in type library in C++ to represent our modified objective function while reusing the Gauss-Newton optimization functionality. The type library exposes properties of the edges including the error metric, reweighted information matrix, and robust Mahalanobis distance function.

A. Datasets

The first dataset is the commonly compared landmark-based dataset Victoria Park released with iSAM [4]. We apply different types of simulated corruption to the dataset to generate multiple synthesized datasets to evaluate the effect on different approaches. For the purpose of evaluating performance against moving landmarks, several other commonly used datasets are not suitable because they are pose-only graphs without landmarks. The Victoria Park dataset of 2-D odometry and landmark observations contains 6969 robot poses, 6968 odometry measurements, 151 landmarks, and 3640 landmark measurements obtained.

The second dataset is based on a set of real world data collected in crowded environment at the Alcazar of Seville with a lot of tourists [10]. The original datasets provide wheel odometry, stereo images, and laser scans. In the indoor GPS denied environment, the provided ground truth map and trajectory are built using a non-linear batch optimization-based SLAM method with an approximate accuracy of 20 cm. We extract potentially moving landmarks from a indoor subset of this dataset through a standard pipeline of feature detection and extraction, feature correspondence, and stereo estimation of keypoint depth implemented with OpenCV [2] and ROS [9]. Due to lack of visual detection of loop closure, we also add in a loop closure obtained from laser scans instead of image data at the start-end point. This dataset tests the typical scenario of visual SLAM with wheel odometry. The extracted dataset consists of 4892 robot poses, 5808 measurements, and 13454 measurements.

B. Prior Methods with Moving Landmarks

In the first test, we evaluate the performance of the state-of-the-art robust SLAM method, a implementation of Max-Mixture [7], given dynamic landmark measurements. Max-Mixture is a robust extension to classical graph SLAM using Gaussian mixture in factor representation, that is, $\boldsymbol{x}_i = \sum_c \phi_c \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ for component c and weight ϕ_c, with observations being components, also similarly for $\boldsymbol{z}_i$. Max-Mixture uses a max function to approximate and efficiently evaluate the sum of Gaussians. It is well-known for its capability of handling a large amount of incorrect loop closures, but it is untested against moving landmarks, and this experiment can be representative to other approaches in the robust SLAM literature.

We apply different perturbations to all observations associated with a certain landmark which has the most observations associated in the dataset, and study the different effects of the perturbations on the optimal estimate of the trajectory and the map of all landmarks obtained by Max-Mixture graph SLAM. The Victoria Park dataset contains 2-D odometry and landmark observations. As shown in Figure 3 the left plot is the control group without perturbation, and is accurate to the ground truth. A single southward bias is applied to all observations of the landmark in the middle plot, simulating sensory outliers. A temporally increasing bias is applied to all observations of the landmark in the right plot, simulating a southward moving landmark.

As the middle plot shows, Max-Mixture is still capable of handling noise of large bias or spurious loop closures introduced by simulated sensory fault. It correctly rejects outlier landmark observations and recovers the position of the perturbed landmark. However, it completely fails to

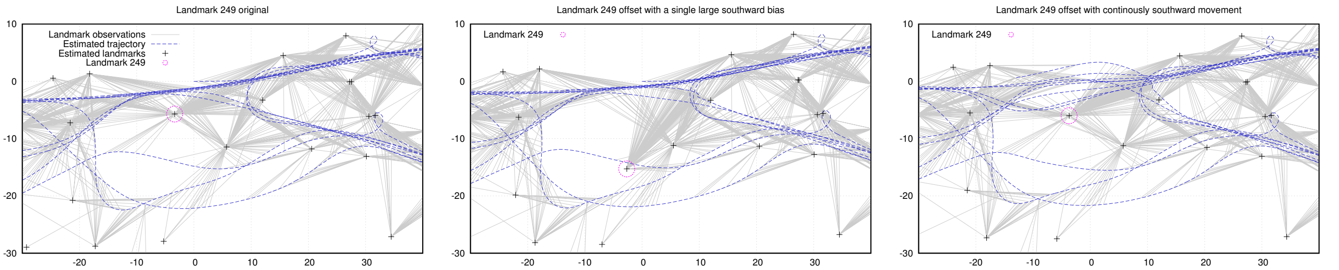


Fig. 3: The optimal estimates of the trajectory and landmark positions obtained by Max-Mixture graph SLAM on the Victoria Park dataset. Left: low convergence uncertainty (measured by objective value). Middle: high convergence uncertainty because of rejection of outlier observations of landmark 249. Right: low convergence uncertainty, no outlier detected by Max-Mixture.

reject any unlikely landmark observations and largely distorts the resulting trajectory when given moving landmark measurements. An explanation for this is that each clique of landmarks with coherent motion forms a plausible reference frame for related observations. Inference based on each independent reference frame will reach plausible estimate of robot trajectory and the map, however robust SLAM methods which assume stationary landmarks will average over a sum of different reference frames and lead to wrong conclusions.

C. Comparison with Prior Methods

To benchmark the robustness of the proposed approach and to show its correctness and feasibility, we use the Victoria Park dataset that has been used in a number of publications before. The dataset consists of pose graphs in 2D and contain several thousand poses and landmark constraints. We corrupted the data by setting landmark “249” (circled in red) in a constant northward movement, where its eventual position is roughly 7 meters north of its original location. We chose landmark “249” due to the relatively large amount of observations associated with this particular landmark in the dataset. We expect the more observations there are on a landmark, the greater its movement would corrupt the final optimization results and the more possible that existing robust SLAM methods would fail.

We compare the results of Dynamic Covariance Scaling robust kernel as discussed in our formulation, and also the SLAM with EM approach proposed in [11]. Figure 4 shows the results for DCS, standard EM, and our approach. As shown in the figure, robust observations alone are unable to handle moving landmarks. Because observation-based robust methods do not characterize the mobility of the landmark, its effect on associated observations and assume independence between observations, which is not the case for moving landmarks. A moving landmark will cause associated observations to be inconsistent in a plausible way such that a subgroup of the observations might be able to converge to a local minimum but the rest of the observations would distort the result elsewhere.

Normal SLAM with EM was proposed in [11] for datasets with very large movement. Their formulation is similar with ours except it is without the robust factor v_k. As shown in Figure 4, the resulting trajectory is distorted. One explanation for this is from the characteristics of the datasets. In their paper, they investigated datasets containing landmarks moved from one room to another room over a long period of time. Such long-duration mobility is not the case in the Victoria Part dataset, where the motion of the landmark is relatively small and continuous. In fact, the motion of the corrupted landmark is doubled and normal EM is able to learn the mobility correctly.

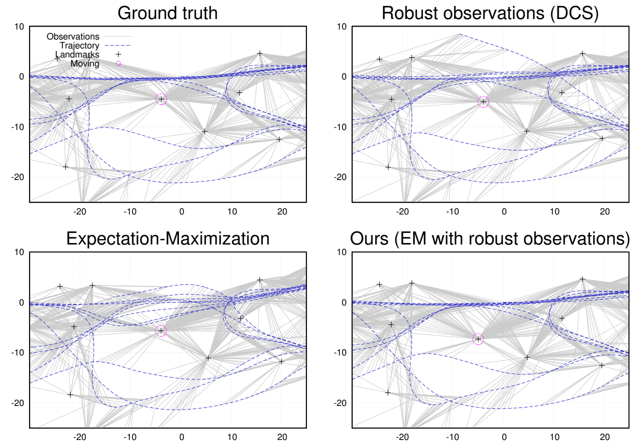


Fig. 4: Dataset corrupted by the landmark with small movement on the Victoria Park dataset

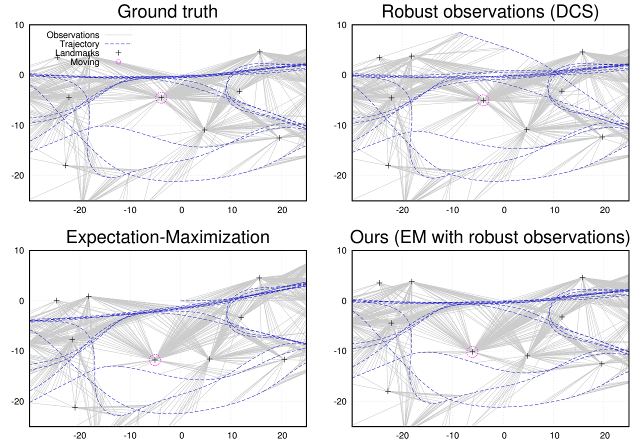


Fig. 5: Dataset corrupted by the landmark with large movement on the Victoria Park dataset

Our robust back-end however, is able to converge to a correct solution in a few iterations. Our result is also verified by examining the learned parameters w_{j_k} and v_k of all landmarks, which is close to zero for the actual outlier landmarks, thus correctly deactivating associated corrupted observations. The improvement is explained by our utilization of the robust kernel to make the EM algorithms reliably learn the mobility indicators of the landmarks correctly identify the actual moving landmarks. Thus our approach is able to obtain the best results in both situations.

In Figures 6 and 7, the result also shows additionally comparison of different methods on the Alcazar of Seville dataset. As can be seen, the trajectory estimation obtained with only robust observations using DCS kernel is significantly distorted. Our method is able to recover the trajectory approximately. The estimated trajectory does not completely converge on the ground truth trajectory due to less than full coverage of extracted landmarks and limited accuracy in depth estimation.

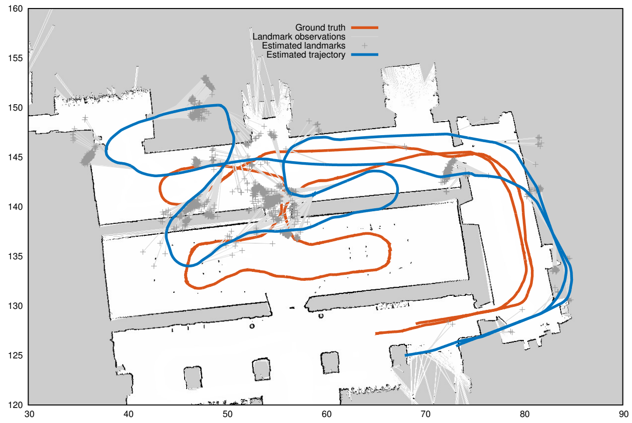


Fig. 6: Estimation with only robust observations (DCS) on the Alcazar of Seville dataset. Background is the ground truth map.

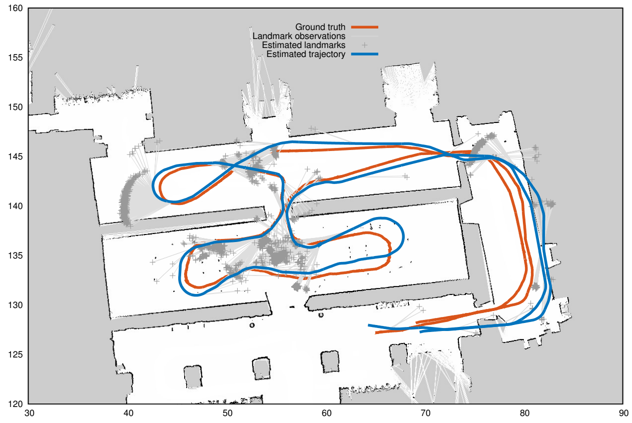


Fig. 7: Estimation with our method on the Alcazar of Seville dataset.

VI. Discussion and Future Work

We have shown the basic efficacy of our approach, but the evaluation is far from being comprehensive. The performance under different percentage of randomly generated outliers, and different sources and types of datasets including 3-D datasets are needed for a complete understanding of the performance in our approach. Due to the different nature of source of errors of moving landmarks than spurious loop closure, the effect of different numbers of observations associated with landmarks also needs to be taken into account for systematic evaluation.

We have examined the theoretical basis for the incremental variant of the EM algorithms. Considering the fast convergence of our batch EM algorithm it might be possible to integrate the incremental algorithm into the graph optimization process without loss of the correctness of the EM algorithms, and making our approach suitable for real-time update.

In our formulation, there are pre-defined parameters λ for penalizing against removing too many landmarks, which is essentially a decision boundary, and Φ for the robust kernel. The appropriate method to choose the parameters and their sensitivity to environmental factors and dataset characteristics need to be examined. There are multiple robust SLAM methods proposing different penalty terms which derive into these parameters. A comparison of the alternative choices of penalty terms is also necessary.

VII. Conclusion

This paper proposed a new method to characterize and identify the mobility of potentially moving landmarks by using the EM algorithms with robust kernel. The feasibility of the proposed approach was shown and evaluated on synthesized datasets from a standard dataset. We compared with existing state-of-the-art methods and showed that observation based robust methods are unable to handle moving landmarks while EM alone without robust kernel does not deal with small continuous movement robustly. We also propose an incremental variant of our approach which will be suitable for real-time incremental update in future implementation.

Appendix

A. SLAM as a Least Squares Problem

For completeness of the paper, we review how to form the SLAM optimization problem as a least squares problem, following the derivation in [4]. Recall that in equation 7, we need to solve a quadratic program, however we may approximate the terms with first order polynomials. For the process term,

$$\begin{aligned} &f_i(\boldsymbol{x}_{i-1}, \boldsymbol{u}_i) - \boldsymbol{x}_i \\ &\qquad \approx f_i(\boldsymbol{x}^0_{i-1}, \boldsymbol{u}_i) + \boldsymbol{F}_i^{i-1}\delta\boldsymbol{x}_{i-1} - (\boldsymbol{x}^0_i + \delta\boldsymbol{x}_i) \\ &\qquad = \boldsymbol{F}_i^{i-1}\delta\boldsymbol{x}_{i-1} + \boldsymbol{G}_i^i\delta\boldsymbol{x}_i - \boldsymbol{a}_i \end{aligned} \tag{11}$$

where $\boldsymbol{I}$ is the identity and

$$\begin{aligned} \boldsymbol{F}_i^{i-1} &= \frac{\partial f_i(\boldsymbol{x}_{i-1}, \boldsymbol{u}_i)}{\partial \boldsymbol{x}_{i-1}}\Big|_{\boldsymbol{x}^0_{i-1}}, \\ \boldsymbol{G}_i^i &= -\boldsymbol{I}, \text{(Identity matrix)} \\ \boldsymbol{a}_i &= \boldsymbol{x}^0_i - f_i(\boldsymbol{x}^0_{i-1}, \boldsymbol{u}_i). \end{aligned}$$

For the measurement term:

$$
\begin{aligned}
h_k(\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k}) - \boldsymbol{z}_k & \\
&\approx h_k(\boldsymbol{x}^0_{i_k}, \boldsymbol{l}^0_{j_k}) + \boldsymbol{H}^{i_k}_k \delta\boldsymbol{x}_{i_k} + \boldsymbol{J}^{j_k}_k \delta\boldsymbol{l}_{j_k} - \boldsymbol{z}_k \\
&= \boldsymbol{H}^{i_k}_k \delta\boldsymbol{x}_{i_k} + \boldsymbol{J}^{j_k}_k \delta\boldsymbol{l}_{j_k} - \boldsymbol{c}_k
\end{aligned} \tag{12}
$$

where,

$$
\begin{aligned}
\boldsymbol{H}^{i_k}_k &= \frac{\partial h_k(\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k})}{\partial \boldsymbol{x}_{i_k}}|_{(\boldsymbol{x}^0_{i_k}, \boldsymbol{l}^0_{j_k})}, \\
\boldsymbol{J}^{j_k}_k &= \frac{\partial h_k(\boldsymbol{x}_{i_k}, \boldsymbol{l}_{j_k})}{\partial \boldsymbol{l}_{j_k}}|_{(\boldsymbol{x}^0_{i_k}, \boldsymbol{l}^0_{j_k})}, \\
\boldsymbol{c}_k &= \boldsymbol{z}_k - h_k(\boldsymbol{x}^0_{i_k}, \boldsymbol{l}^0_{j_k})
\end{aligned}
$$

Therefore, the optimization problem thus becomes

$$
\begin{aligned}
\delta\boldsymbol{\theta}^* = \arg\min\nolimits_{\delta\boldsymbol{\theta}} \{ & \sum_{i=1}^{M} ||\boldsymbol{F}^{i-1}_i \delta\boldsymbol{x}_{i-1} + \boldsymbol{G}^i_i \delta\boldsymbol{x}_i - \boldsymbol{a}_i||^2_{\boldsymbol{\Gamma}_i} \\
& + \sum_{k=1}^{K} w_{j_k} ||v_k(\boldsymbol{H}^{i_k}_k \delta\boldsymbol{x}_{i_k} + \boldsymbol{J}^{j_k}_k \delta\boldsymbol{l}_{j_k} - \boldsymbol{c}_k)||^2_{\boldsymbol{\Sigma}_k} \}
\end{aligned} \tag{13}
$$

To rewrite the Mahalanobis norm, notice that

$$
||\boldsymbol{e}||^2_{\boldsymbol{\Gamma}} = \boldsymbol{e}^T \boldsymbol{\Gamma}^{-1} \boldsymbol{e} = ||\boldsymbol{\Gamma}^{-T/2} \boldsymbol{e}||^2 \tag{14}
$$

Therefore, we can collect all the Jacobian matrices multiplied by either $\boldsymbol{\Gamma}_i^{-T/2}$ or $v_k w_{j_k}^{1/2} \boldsymbol{\Sigma}_k^{-T/2}$, specifically,

$$
\begin{aligned}
\delta\boldsymbol{\theta}^* = & \arg\min_{\delta\boldsymbol{\theta}} \{ \\
& \sum_{i=1}^{M} ||\boldsymbol{\Gamma}_i^{-T/2} \boldsymbol{F}^{i-1}_i \delta\boldsymbol{x}_{i-1} + \boldsymbol{\Gamma}_i^{-T/2} \boldsymbol{G}^i_i \delta\boldsymbol{x}_i - \boldsymbol{\Gamma}_i^{-T/2} \boldsymbol{a}_i||^2 \\
& + \sum_{k=1}^{K} ||v_k \sqrt{w_{j_k}} \boldsymbol{\Sigma}_k^{-T/2} \boldsymbol{H}^{i_k}_k \delta\boldsymbol{x}_{i_k} \\
& \quad + v_k \sqrt{w_{j_k}} \boldsymbol{\Sigma}_k^{-T/2} \boldsymbol{J}^{j_k}_k \delta\boldsymbol{l}_{j_k} - v_k \sqrt{w_{j_k}} \boldsymbol{\Sigma}_k^{-T/2} \boldsymbol{c}_k||^2 \}
\end{aligned} \tag{15}
$$

We concatenate the matrices $\boldsymbol{F}, \boldsymbol{G}, \boldsymbol{H}, \boldsymbol{J}$ in block form into a square matrix $\boldsymbol{A}$, $\delta\boldsymbol{x}, \delta\boldsymbol{l}$ into $\delta\boldsymbol{\theta}$, and $\boldsymbol{a}$ and $\boldsymbol{c}$ into the residual vector $\boldsymbol{b}$ to then obtain the standard least squares problem:

$$
\delta\boldsymbol{\theta}^* = \arg\min_{\delta\boldsymbol{\theta}} ||\boldsymbol{A}\delta\boldsymbol{\theta} - \boldsymbol{b}||^2 \tag{16}
$$

References

[1] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, "Robust map optimization using dynamic covariance scaling," in *Proc. of IEEE Conf. on Robotics and Automation*, 2013.

[2] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[3] D. Hähnel, W. Burgard, D. Fox, and S. Thrun, "A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements," in *Proc. of the IEEE Conf. on Intelligent Robots and Systems*, 2003.

[4] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. on Robotics (TRO)*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[5] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of IEEE Conf. on Robotics and Automation*. IEEE, 2011, pp. 3607–3613.

[6] R. M. Neal and G. E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in *Learning in graphical models*. Springer, 1998, pp. 355–368.

[7] E. Olson and P. Agarwal, "Inference on networks of mixtures for robust robot mapping," in *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[8] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2262–2269.

[9] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[10] R. Ramón-Vigo, J. Pérez, F. Caballero, and L. Merino, "Navigating among people in crowded environment: Datasets for localization and human robot interaction," in *Proceedings of the Workshop on Robots in Clutter: Perception and Interaction in Clutter, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[11] J. G. Rogers, A. J. Trevor, C. Nieto-Granda, and H. I. Christensen, "Slam with expectation maximization for moveable object tracking," in *Proc. of IEEE Conf. on Intelligent Robots and Systems*. IEEE, 2010, pp. 2077–2082.

[12] N. Sunderhauf and P. Protzel, "Switchable constraints for robust pose graph slam," in *Proc. of IEEE Conf. on Intelligent Robots and Systems*, 2012.

[13] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas," in *Proc. of IEEE Conf. on Robotics and Automation*, vol. 1. IEEE, 2003, pp. 842–849.

[14] D. F. Wolf and G. S. Sukhatme, "Mobile robot simultaneous localization and mapping in dynamic environments," *Autonomous Robots*, vol. 19, pp. 53–65, July 2005.

[15] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," Mitsubishi Electric Research Laboratories, Tech. Rep. TR2001-22, 2001.